CLAIMS:

What is claimed is:

1. A method in a data processing system for processing a Java server page, the method comprising:

translating the Java server page into a document object model object;

configuring a set of visitor classes for invocation in a selected sequence; and

processing the document object model using the set of visitor classes in the selected sequence to perform a desired set of custom functions on the document object model.

2. The method of claim 1 further comprising:

validating syntax in the Java server page.

3. The method of claim 1, wherein the set of visitor classes for invocation in the selected sequence is defined in a configuration file.

4. The method of claim 3, wherein the configuration file is an extensible markup language file.

5. The method of claim 3, wherein the selected sequence is defined in the configuration file.

6. The method of claim 1, wherein

the document object model object includes a set of nodes and wherein the processing step includes:

invoking methods in the set of visitor classes on each node in the set of nodes in the selected sequence.

7. The method of claim 1 further comprising:

storing results, as processing the document object model object occurs by selected method in the methods, in a hash map, wherein the results in the hash map are used by subsequently invoked methods.

8. The method of claim 1, wherein the java server page is translated into a document object model object using a document object model generator.

9. The method of claim 2, wherein the Java server page is validated using a Java server page translator.

10. The method of claim 9, wherein the Java server page translator invokes a visitor class to validate elements in the document object model object against a syntax for a Java server page specification.

11. The method of claim results from processing by a first visitor class in the set of visitor classes are passed to a second visitor class in the set of visitor classes.

12. A data processing system for processing a Java / server page, the data processing system comprising:

translating means for translating the Java server page into a document object model object;

configuring means for configuring a set of visitor classes for invocation in a selected sequence; and

processing means for processing the document object model using the set of visitor classes in the selected

sequence to perform a desired set of custom functions on the document object model.

13. The data processing system of claim 12 further comprising:

validating means for validating syntax in the Java server page.

14. The data processing system of claim 12, wherein the set of visitor classes for invocation in the selected sequence is defined in a configuration file.

15. The data processing system of claim 14, wherein the configuration file is an extensible markup language file.

16. The data processing system of claim 14, wherein the selected sequence is defined in the configuration file.

17. The data processing system of claim 12, wherein the document object model object includes a set of nodes and wherein the processing means includes:

means for invoking methods in the set of visitor classes on each node in the set of nodes in the selected sequence.

18. The data processing system of claim 12 further comprising:

storing means for storing results, as processing the document object model object occurs by selected method in the methods, in a hash map, wherein the results in the hash map are used by subsequently invoked methods.

19. The data processing system of claim 12, wherein the java server page is translated into a document object model object using a document object model generator.

20. The data processing system of claim 13, wherein the Java server page is validated using a Java server page translator.

21. The data processing system of claim 20, wherein the Java server page translator invokes a visitor class to validate elements in the document object model object against a syntax for a Java server page specification.

22. The data processing system of claim 1, wherein results from processing by a first visitor class in the set of visitor classes are passed to a second visitor class in the set of visitor classes.

23. A computer program product in computer readable medium for processing a Java server page, the computer program product comprising:

first instructions for translating the Java server page into a document object model object;

second instructions for configuring a set of visitor classes for invocation in a selected sequence; and

third instructions for processing the document object model using the set of visitor classes in the selected sequence to perform a desired set of custom functions on the document object model.

24. The computer program product of claim 23 further comprising:

fourth instructions for validating syntax in the Java server page.

25. The computer program product of claim 23, wherein the set of visitor classes for invocation in the selected sequence is defined in a configuration file.

26. The computer program product of claim 25, wherein the configuration file is an extensible markup language file.

27. The computer program product of claim 25, wherein the selected sequence is defined in the configuration file.

28. The computer program product of claim 23, wherein the document object model object includes a set of nodes and wherein the third instructions includes:

sub instructions for invoking methods in the set of visitor classes on each node in the set of nodes in the selected sequence.

29. The computer program product of claim 23 further comprising:

fourth instructions for storing results, as processing the document object model object occurs by selected method in the methods, in a hash map, wherein the results in the hash map are used by subsequently invoked methods.

30. The computer program product of claim 23, wherein the java server page is translated into a document object model object using a document object model generator.

31. The computer program product of claim 24, wherein the Java server page is validated using a Java server page translator.

32. The computer program product of claim 31, wherein the Java server page translator invokes a visitor class to validate elements in the document object model object against a syntax for a Java server page specification.

33. The computer program product of claim 23, wherein results from processing by a first visitor class in the set of visitor classes are passed to a second visitor class in the set of visitor classes.

34. A data processing system for processing a Java server page, the data processing system comprising:

a bus system;

a memory connected to the bus system, wherein the memory includes a set of instructions;

a processing unit connected to the bus system, wherein the processing unit executes the set of instructions to translate the Java server page into a document object model object; configure a set of visitor classes for invocation in a selected sequence; and process the document object model using the set of visitor classes in the selected sequence to perform a

desired set of custom functions on the document object model.